ENCRYPTING RECEIVED CONTENT

FIELD OF THE INVENTION

This invention relates generally to broadband communications systems, such as subscriber television systems, and more specifically to encrypting content at a receiver in a broadband communication system.

BACKGROUND OF THE INVENTION

Frequently, broadband systems transmit television signals and programs to subscribers of a conditional access system. Broadband systems, such as cable and satellite television systems, typically include a headend for receiving programming and/or data from various sources and redistributing the programming and other data through a distribution system to subscribers. The headend receives programming signals from a variety of sources, combines the programming signals from the various sources, and transmits the combined signals through the distribution system to subscriber equipment. The distribution system can include a variety of media, such as coaxial cable, fiber optic cable, and satellite links among others. In a subscriber television system, the subscriber equipment, which receives the signals from the headend, can include, among others, a cable-ready television, a cable-ready video cassette recorder (VCR), or a digital subscriber communications terminal (DSCT) that is connected to a television, computer, or other display device.

The headend uses modulators to control the streams of data into the distribution system. Increasingly, the headend is receiving and transmitting programming in a digital format, for example, Moving Pictures Expert Group (MPEG) format, instead of an analog format. Transmitting programs in MPEG format is advantageous because multiple digitized programs can be combined and transmitted in, for example, 6 MHz of bandwidth, which is the same amount of bandwidth that is required to transmit a single analog channel or program, and in comparison to analog programs, MPEG or digitized programs provide a cleaner and sharper image and sound. Various error correction schemes enable the digital packets to be transmitted through a digital network with minimal distortion or error.

In order to thwart unauthorized access to the content of the broadband system, the content is usually encrypted at the headend prior to distribution. The headend provides the authorized subscribers of the broadband system with the keys necessary to decrypt the encrypted content. Typically, content such as programs or instances of service are encrypted using a symmetrical cryptographic algorithm. A symmetrical cryptographic algorithm uses a pair of functions (F and F^{-1}) and a single key (k) or keys for both encryption and decryption. When a function, F or F^{-1}, is applied to cleartext (C) using a key (k), the cleartext is converted to ciphertext (C′). The produced ciphertext, C′, depends upon the cleartext, C, the key, k, and upon which function, F or F^{-1}, was used to produce it. For example:

$$F(k; C) = C'\{F; k\},$$
$$F^{-1}(k; C) = C'\{F; k\}, \text{ and}$$
$$C'\{F; k\} \neq C' \{F^{-1}; k\}.$$

Similarly, ciphertext produced from the same cleartext and the same function, G = F or F^{-1}, but with different keys k_1 and k_2, are different:

$$G(k_1; C) \neq G(k_2; C).$$

Ciphertext can be converted back into cleartext by using the appropriate function with the appropriate key. The appropriate function being the function that is the inverse of the function used for generating the ciphertext. Either function can be used for converting cleartext to ciphertext, and when the same key is used, each reverses the operation performed by the other, e.g.,

$$F(k; F^{-1} (k; C)) = F^{-1}(k; F (k; C)) = C.$$

The function F is conventionally referred to as the encryption function (E) and the function F^{-1} is conventionally referred to as the decryption function (D). However, this conventional naming scheme can be confusing, because the decryption function can be used for encryption, i.e., converting cleartext to ciphertext, and the encryption function can be used for decryption, i.e., converting ciphertext to cleartext.

Some cryptographic algorithms such as 3DES use multiple functions and keys to convert between cleartext and ciphertext. In one embodiment of 3DES encryption, cleartext is converted to ciphertext according to the following scheme:

$$F(k_3; F^{-1} (k_2; F (k_1; C))) = C'''\{F, F^{-1}, F; k_1, k_2, k_3\}.$$

The first operation, $F(k_1; C)$, produces ciphertext $C'(F; k_1)$. The single prime (') designates that the ciphertext $C'\{F; k_1\}$ has one layer of encryption thereon. The second operation, $F^{-1}(k_2; F(k_1; C))$ or $F^{-1}(k_2; C'\{F; k_1\})$, applies the function F^{-1} using the k_2 on the ciphertext $C'\{F; k_1\}$ to produce the ciphertext $C''\{F, F^{-1}; k_1, k_2\}$, thereby applying a second layer of encryption. The second operation does not produce cleartext, C, because different keys are used, e.g., k_1 does not equal k_2. The third operation, which adds a third layer of encryption, applies the function F using the key k_3 on the $C''\{F, F^{-1}; k_1, k_2\}$ to produce the ciphertext $C'''\{F, F^{-1}, F; k_1, k_2, k_3\}$. To convert $C'''\{F, F^{-1}, F; k_1, k_2, k_3\}$ back to cleartext C, the inverse functions F and F^{-1} must be applied with the appropriate key in reverse order:

$$F^{-1}(k_1; F(k_2; F^{-1}(k_3; C'''\{F, F^{-1}, F; k_1, k_2, k_3\}))) = C.$$

In an alternative implementation of the 3DES cryptographic algorithm, the first key and third keys are the same, $k_1 = k_3$. In that case, cleartext is converted to ciphertext according to the following scheme:

$$F(k_1; F^{-1}(k_2; F(k_1; C))) = C'''\{F, F^{-1}, F; k_1, k_2, k_1\}.$$

In theory, the packets of a digital program can be reproduced or copied without error. Thus, a subscriber of a subscriber network who receives a digital program can record the program and copy it, and the copy will be virtually identical to the original. Therefore, there exists concern about illegal copying or bootlegging of digital content. The operators of a subscriber network and the content providers want to provide the subscribers of the digital network with the programming and services desired by the subscribers, but the digital content owners want to prevent the subscribers from making and distributing bootleg copies of the digitized programs and services. Thus, there exists a need for an apparatus that protects the property interests of the digital content owners, while providing the subscribers with the desired digital content.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram representation of an example of a broadband communications system, such as a cable television system, in which the embodiments of the present invention may be employed.

FIG. 2 is a block diagram representation of an example of a headend in the broadband communication system in which embodiments of the present invention may be employed.

FIG. 3 is a block diagram representation of an example of a transactional encryption device.

FIG. 4 is a block diagram representation of an MPEG transport packet.

FIG. 5 is a block diagram representation of an example that illustrates the levels of security in the example broadband communication system at the example headend.

FIG. 6 is a block diagram representation of an example of the digital subscriber communication terminal.

FIG. 7, illustrates the levels of security in the broadband communication system at the DSCT.

FIG. 8 is a flowchart representation of an example for storing a program.

FIG. 9 is a block diagram representation of an example of the processing of a program at the headend and the DSCT.

FIG. 10 is a block diagram representation of another example of the processing of a program at the headend and the DSCT.

FIG. 11 is a block diagram representation of another example of the processing of a program at the headend and the DSCT.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Embodiments of the present invention will be described more fully hereinafter with reference to the accompanying drawings in which like numerals represent like elements throughout the several figures, and in which an exemplary embodiment of the invention is shown. The present invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. The examples set forth herein are non-limiting examples and are merely examples among other possible examples.

The logic of the present invention can be implemented in hardware, software, firmware, or a combination thereof. In the preferred embodiment(s), the logic is implemented in software or firmware that is stored in a memory and that is executed by a suitable instruction execution system. If implemented in hardware, as in an alternative embodiment, the logic can be implemented with any or a combination of the following technologies, which are all well known in the art: a discrete logic circuit(s) having logic gates for implementing logic functions upon data signals, an application specific integrated circuit (ASIC) having appropriate combinational logic gates, a programmable gate array(s) (PGA), a field programmable gate array (FPGA), *etc.*

Any process descriptions or blocks in flow charts should be understood as representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process, and alternate implementations are included within the scope of the preferred embodiment of the present invention in which functions may be executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those reasonably skilled in the art of the present invention.

Television System Overview

The preferred embodiment of the invention is best understood within the context of a two-way, interactive subscriber television system or a subscriber network, as an example. In this discussion, the two-way interactive digital subscriber television system is also referred to as a Digital Broadband Delivery System (DBDS). An overview of an exemplary DBDS is provided in U.S. Patent No. 6,157,719, entitled "*Conditional Access System*", which is hereby incorporated by reference in its entirety. A function of the DBDS is to provide interfaces to content and service providers, entitlement agents, control access to and the use of the content and services, and to distribute the content and services to subscribers. The DBDS uses Motion Picture Experts Group (MPEG) transport streams for delivery of video, audio, and digitized data entertainment services. MPEG as referenced in this application is described in the MPEG-1 and MPEG-2 standards. The MPEG-1 standards (ISO/IEC 11172) and the MPEG-2 standards (ISO/IEC 13818) are described in detail in the International Organization for Standardization document ISO/IEC JTC1/SC29/WG11 N (June 1996 for MPEG-1 and July 1996 for MPEG-2), which is hereby incorporated by reference in its entirety. The content and services distributed to the subscribers can include programming and services such as local television channels, premium movie channels, video-on-demand (VOD), telephone services, Internet access, and audio programming, among others.

Referring to FIG. 1, a digital broadband distribution system (DBDS) 100 includes, in one example among others, a headend 102, a plurality of hubs 104, multiple nodes 106, a plurality of subscriber locations 108, and a plurality of digital subscriber communication terminals (DSCTs) 110. The headend 102 provides the interface between the DBDS 100 and content and service providers 114, such as broadcasters, internet service providers, and the like via communication link 162. The transmission medium

162 between the headend 102 and the content and service providers 114 can be two-way. This allows for two-way interactive services such as Internet access via DBDS 100, video-on-demand, interactive program guides, etc. In the preferred embodiment, the hubs 104 are also in direct two-way communication with the content and service providers 114 via communication link 162 for providing two-way interactive services.

In the preferred embodiment, the headend 102 is in direct communication with the hubs 104 via communication link 150. In addition, the headend 102 can be in direct communication with some or all of the nodes 106 via communication link 152 or in direct communication with some or all of the subscriber locations 108 via communication link 154. Whether the headend 102 communicates directly with nodes 106 and/or subscriber locations 108 is a matter of implementation. The hub 104 receives programming and other information from headend 102 via transmission medium 150, typically in an ethernet medium and transmits information and programming via transmission medium 152 to nodes 106, which then transmit the information to subscriber locations 108 through transmission medium 154. Again, whether the hub 104 communicates directly to subscriber locations 108 or to nodes 106 is matter of implementation, and in the preferred embodiment, the hub 104 is also adapted to transmit information and programming directly to subscriber locations 108 via transmission medium 154.

In the preferred embodiment, the transmission medium 150 and 152 are optical fibers that allow the distribution of high quality and high-speed signals, and the transmission medium 154 is either broadband coaxial cable or optical fiber. In alternative embodiments, the transmission media 150, 152 and 154 can incorporate one or more of a variety of media, such as optical fiber, coaxial cable, and hybrid fiber-coax (HFC), satellite, direct broadcast, or other transmission media known to those skilled in the art. Typically, the transmission media 150, 152 and 154 are two-way communication media through which both in-band and out-of-band information are transmitted. Through the transmission media 150, 152 and 154 subscriber locations 108 are in direct or indirect two-way communication with the headend 102 and/or the hub 104.

The hub 104 functions as a mini-headend for the introduction of programming and services to sub-distribution network 160. The sub-distribution network 160 includes hub 104 and the plurality of nodes 106 connected to hub 104. Having a plurality of hubs 104 that function as mini-headends facilitates the introduction of different programming, data and services to different sub-distribution networks of DBDS 100. For example, the subscriber location 108(b), which is connected to node 106(b), can have different

services, data and programming available than the services, data and programming available to subscriber location 108(c), which is connected directly to headend 102, even though the subscriber locations 108(b) and 108(c) may be in close physical proximity to each other. Services, data and programming for subscriber location 108(b) are routed through hub 104 and node 106(b); and hub 104 can introduce services, data and programming into the DBDS 100 that are not available through the headend 102.

At the subscriber locations 108 a decoder or a DSCT 110 provides the two-way interface between the DBDS 100 and the subscriber. The DSCT 110 decodes and further process the signals for display on a display device, such as a television set (TV) 112 or a computer monitor, among other examples. Those skilled in the art will appreciate that in alternative embodiments the equipment for decoding and further processing the signal can be located in a variety of equipment, including, but not limited to, a DSCT, a computer, a TV, a monitor, or an MPEG decoder, among others.

As will be explained in detail hereinbelow, secure communication between the headend 102 and the DSCTs 110 is accomplished using pairs of asymmetrical keys known to those skilled in the art, such as Rivest, Shamir, & Adleman (RSA) public key encryption technology. Briefly described, an asymmetrical key pair includes a public key, which is distributed to the public, and a private key, which is not distributed. Content that is encrypted with a public key can only be decrypted using the corresponding private key. A message that is signed with a private key is authenticated with the corresponding public key. Thus, after headend 102 and the DSCT 110 have exchanged public keys they can securely communicate. The content of a message for the particular DSCT 110 is encrypted using the public key of the particular DSCT 110, and only the particular DSCT 110 that has the corresponding private key can decrypt the content of the message. The message can also be signed by the private key of the headend 102, and in that case the DSCT 110 uses the public key of the headend 102 to authenticate the message. For details regarding cryptography that a reasonably skilled person would understand see, Bruce Schneier, *"Applied Cryptography"*, John Wiley & Sons, 1994.

Headend

Referring to FIG. 2, in a typical system of the preferred embodiment of the invention, the headend 102 receives content from a variety of input sources, which can include, but are not limited to, a direct feed source (not shown), a video camera (not shown), an application server (not shown), and other input sources (not shown). The input signals are transmitted from the content providers 114 to the headend 102 via a variety of communication links 162, which include, but are not limited to, satellites (not shown), terrestrial broadcast transmitters (not shown) and antennas (not shown), and direct lines (not shown). The signals provided by the content providers, or entitlement agents, can include a single program or a multiplex that includes several programs, and typically, a portion of the content from the input sources is encrypted.

The headend 102 generally includes a plurality of receivers 218 that are each associated with a content source. Generally, the content is transmitted from the receivers 218 in the form of transport stream 240. MPEG encoders, such as encoder 220, are included for digitally encoding things such as local programming or a feed from a video camera. Typically, the encoder 220 produces a variable bit rate transport stream. Some of the signals may require additional processing, such as signal multiplexing prior to being modulated. Such multiplexing is done by multiplexer 222.

A switch, such as asynchronous transfer mode (ATM) switch 224, provides an interface to an application server (not shown). There can be multiple application servers providing a variety of services such as, among others, a data service, an Internet service, a network system, or a telephone system. Service and content providers 114 (shown in FIG. 1) may download content to an application server located within the DBDS 100 or in communication with DBDS 100. The application server may be located within headend 102 or elsewhere within DBDS 100, such as in a hub 104.

Typically, the headend 102 includes a server such as a video-on-demand (VOD) pump 226. VOD pump 226 provides video and audio programming such as VOD pay-per-view programming to subscribers of the DBDS 100. Usually, the content from VOD pump 226 is provided in the form of transport stream 240.

The various inputs into the headend 102 are then combined with the other information, which is specific to the DBDS 100, such as local programming and control information. The headend 102 includes a multi-transport stream receiver-transmitter 228 that receives a plurality of transport streams 240 and transmits a plurality of transport streams 242. In the preferred embodiment, the multi-transport stream receiver-

transmitter 228 includes a plurality of modulators, such as, but not limited to, Quadrature Amplitude Modulation (QAM) modulators, that convert the received transport streams 240 into modulated output signals suitable for transmission over transmission medium 280.

The output signals 242 from the multi-transport stream receiver-transmitters 228 are combined, using equipment such as a combiner 230, for input into the transmission medium 150, and the combined signals are sent via the in-band delivery path 254 to subscriber locations 108. It is to be understood that modulating the output signals 242 is a matter of implementation based at least in part on the transmission medium 280 that carries output signals 242.

In the preferred embodiment, the multi-transport stream receiver-transmitter 228 receives a plurality of input transport streams 240, which include programs, or sessions, and outputs a plurality of radio frequency modulated transport streams 242. In the DBDS 100, video, audio, and control information are encoded as program streams, which are then multiplexed to form transport streams 240. Each output transport stream from multi-transport stream receiver-transmitter 228 is modulated to a set frequency. For the DSCT 110 (shown in FIG. 1) to receive a television program, in the preferred embodiment, among others, the DSCT 110 tunes to the frequency associated with the modulated transport stream that contains the desired information, de-multiplexes the transport stream, and decodes the appropriate program streams.

A system controller, such as control system 232, which preferably includes computer hardware and software providing the functions discussed herein, allows the DBDS system operator to control and monitor the functions and performance of the DBDS 100. The control system 232 interfaces with various components, via communication link 270, in order to monitor and/or control a variety of functions, including the channel lineup of the programming for the DBDS 100, billing for each subscriber, and conditional access for the content distributed to subscribers. Control system 232 provides input to the multi-transport stream receiver-transmitter 228 for setting its operating parameters, such as system specific MPEG table packet organization or conditional access information.

Control information and other data can be communicated to DSCTs 110 via the in-band delivery path 254 or to DSCTs 110 connected to the headend 102 via an out-of-band delivery path 256. The out-of-band data is transmitted via the out-of-band downstream path 258 of transmission medium 154 by means such as, but not limited to, a

Quadrature Phase-Shift Keying (QPSK) modem array 260, an array of data-over-cable service interface specification (DOCSIS) modems, or other means known to those skilled in the art. Two-way communication utilizes the upstream portion 262 of the out-of-band delivery system. DSCTs 110 transmit out-of-band data through the transmission medium 154, and the out-of-band data is received in headend 102 via out-of-band upstream paths 262. The out-of-band data is routed through router 264 to an application server or to the VOD pump 226 or to control system 232. Out-of-band control information includes such information as a pay-per-view purchase instruction and a pause viewing command from the subscriber location 108 (shown in FIG. 1) to a video-on-demand type application server, and other commands for establishing and controlling sessions, such as a Personal Television session, etc. The QPSK modem array 260 is also coupled to communication link 152 (FIG. 1) for two-way communication with the DSCTs 110 coupled to nodes 106.

The router 264 is used for communicating with the hub 104 through transmission medium 150. Typically, command and control information among other information between the headend 102 and the hub 104 are communicated through transmission medium 150 using a protocol such as but not limited to Internet Protocol. The IP traffic 272 between the headend 102 and hub 104 can include information to and from DSCTs 110 connected to hub 104.

The control system 232, such as Scientific-Atlanta's Digital Network Control System (DNCS), as one acceptable example among others, also monitors, controls, and coordinates all communications in the subscriber television system, including video, audio, and data. The control system 232 can be located at headend 102 or remotely.

In the preferred embodiment, the multi-transport stream receiver-transmitter 228 is adapted to encrypt content prior to modulating and transmitting the content. Typically, the content is encrypted using a cryptographic algorithm such as the Data Encryption Standard (DES) or triple DES (3DES), Digital Video Broadcasting (DVB) Common Scrambling or other cryptographic algorithms or techniques known to those skilled in the art. The multi-transport stream receiver-transmitter 228 receives instructions from the control system 232 regarding the processing of programs included in the input transport streams 240. Sometimes the input transport streams 240 include programs that are not transmitted downstream, and in that case the control system 232 instructs the multi-transport stream receiver-transmitter 240 to filter out those programs. Based upon the instructions received from the control system 232, the multi-transport stream receiver-

transmitter 228 encrypts some or all of the programs included in the input transport streams 240 and then includes the encrypted programs in the output transport streams 242. Some of the programs included in input transport stream 240 do not need to be encrypted, and in that case the control system 232 instructs the multi-transport stream transmitter-receiver 228 to transmit those programs without encryption. The multi-transport stream receiver-transmitter 228 sends the DSCTs 110 the keys that are needed to decrypt the encrypted program. It is to be understood that for the purposes of this disclosure a "program" extends beyond a conventional television program and that it includes video, audio, video-audio programming and other forms of services and digitized content. "Entitled" DSCTs 110 are allowed to use the keys to decrypt encrypted content, details of which are provided hereinbelow.

In the preferred embodiment, the multi-transport stream receiver-transmitter 228 is also adapted to decrypt encrypted content or ciphertext. For the purposes of this disclosure, ciphertext refers to encrypted content, without regard to the source of the content. In other words, the content can be text, video, audio, or any source of content. Sometimes the content provided by the content providers 114 is encrypted, and the multi-transport stream receiver-transmitter 228 applies an encryption function of a cryptographic algorithm to the ciphertext to convert the ciphertext to a different ciphertext, and other times it converts the ciphertext to cleartext, i.e., unencrypted content. As with ciphertext, cleartext can be text, video, audio, or any source of information or content. In the context of this disclosure, cleartext is used to refer to non-encrypted content, not to the type of content.

In the preferred embodiment, the hub 104, which functions as a mini-headend, includes many or all of the same components as the headend 102. The hub 104 is adapted to receive the transport-streams 242 included in the in-band path 254 and redistribute the content therein throughout its sub-distribution network 160. The hub 104 includes a QPSK modem array (not shown) that is coupled to communication links 152 and 154 for two-way communication with DSCTs 110 that are coupled to its sub-distribution network 160.Thus, it is also adapted to communicate with the DSCTs 110 that are coupled to its sub-distribution network 160, with the headend 102, and with the content providers 114.

Referring to FIG. 3, the control system 232 includes, among other components, transaction encryption devices (TEDs) 302, a conditional access authority (CAA) 312, an Entitlement Management Message (EMM) generator 320 and a CAA/TED database 322.

The CAA/TED database 322 includes the public keys and the serial numbers of the DSCTs 110 within the DBDS 100. Each DSCT 110 in the DBDS 100 has a unique serial number, and the serial number, which can be the IP address of the DSCT 110, is used for addressing messages to the DSCT 110. The public key of each DSCT 110 and its serial number are copied when the DSCT 110 is manufactured. In the preferred embodiment, the manufacturer provides a copy of the public key and the serial number of each DSCT 110 to the control system 232. In that case, the manufacturer is a key certification authority that certifies to the operator of the DBDS 100 that a given public key belongs to a specific DSCT 110 (and the control system 232 verifies the certificate to be authentic before placing it into the database of DSCTs 110). In one embodiment, the public keys of the DSCTs 110 are included in messages transmitted from the DSCTs 110 to the control system 232, and the public keys are then stored to the CAA/TED database 322, and the control system 232 verifies the certificate to be authentic before placing it into the database of DSCTs 110.

In the preferred embodiment, the CAA/TED database 322 includes a per-TED database. The per-TED database includes encryption information for each TED 302, such as, but not limited to, current information regarding keys used for encrypting content provided to the DSCTs 110 and expiration times for those keys. The CAA/TED database 322 also includes a per-DSCT database, which includes entitlement information for each DSCT 110 in the DBDS 100. The per-DSCT database can also include customer billing information such as the information required to bill a subscriber for a VOD pay-per-view program or instance of service.

The EMM generator 320 generates message templates that are provided to the CAA 312 and the TEDs 302. The CAA 312 and the TEDs 302 fill in the information to create EMMs that are used for, among other things, controlling access to DSCTs 110, establishing a TED 302 with a DSCT 110, providing limited authority for a TED 302 within the DBDS 100, providing entitlements to a DSCT 110 for programs and instances of service associated with a TED 302 and disestablishing a TED 302 with a DSCT 110. Details of EMMs are provided in U.S. Patent No. 6,157,719, entitled "*Conditional Access System*", which is hereby incorporated by reference in its entirety. The EMM generator 320 generates EMM templates for both the CAA 312 and the TEDs 302.

In the preferred embodiment, each TED 302 is associated with an entitlement agent, which provides content to the DBDS 100. Typically, there is one entitlement agent for the DBDS 100. The entitlement agent receives or generates content that is provided to

the DSCTs 110. For example, the content from the entitlement agent can include, but is not limited to, television programming from content owners such as NBC, HBO, CBS, ABC, etc., audio programming, computer programs, audio information from telephone services, and information from the Internet. In the preferred embodiment of the invention, multiple entitlement agents are allowed to provide content to the DSCTs 110 of the DBDS 100. In an alternative embodiment, there is only one TED 302 for the DBDS 100 and the functionality of the CAA 312 and the TED 302 are combined into a single apparatus. In yet another embodiment, the TED 302 and the CAA 312 are fully integrated into the control system 232, such that a central processing unit (not shown) of the control system 232 implements the logic for providing the functionality of the TED 302 and CAA 312.

Each TED 302 is used for, among other things, generating encryption information used by the multi-transport stream receiver-transmitter 228 for encrypting content, programs or instances of service, provided to the DSCTs 110 and generating decryption information used by the DSCTs 110 for decrypting the received programs or instances of service. The TED 302 generates a multi-session key (MSK), which is used as part of the encryption and decryption process.

The EMM generator 320 provides the TED 302 with an EMM template addressed to a specific DSCT 110, and the TED 302 processes the EMM template to include the MSK in the EMM. The information included in the EMM can also be entitlement information for the DSCT 110 for the programs or instances of service provided by the entitlement agent associated with the TED 302. In the preferred embodiment, the EMMs are transmitted via the out-of-band path 256. The TED 302 provides the EMM to the QPSK modem array 260, which then transmits the EMM to the DSCT 110. In an alternative embodiment, the EMMs are transmitted in-band, and in that case the TED 302 provides the EMM to the multi-transport stream receiver-transmitter 228, which transmits the EMM to the DSCT 110. The MSK is also stored to the CAA/TED database 322 and it is provided to the multi-transport stream receiver-transmitter 228. Details of the EMMs and MSKs are provided hereinbelow.

In the preferred embodiment, a particular TED 302 has at least one public key - private key pair and it uses the private key for signing and decrypting messages. In the preferred embodiment, some or all of the DSCTs 110 in DBDS 100 will have the public key of a particular TED 302, and the particular TED 302 will have access to public keys from some or all of the DSCTs 110. The public key of the particular TED 302 is given to

a given DSCT 110 when the particular TED 302 is established with the given DSCT 110. A TED 302 that is established with a DSCT 110 is allowed to provide content or programs or instances of service to the DSCT 110. Establishment of a TED 302 with a DSCT 110 is controlled by the conditional access authority (CAA) 312 of system controller 232.

The TED 302 includes a central processing unit (CPU) 304, a cryptographic accelerator 318, and a memory 306, which has the private key (or private keys) of the TED 302 stored therein. The memory 306 also includes the logic used for performing the functions of the TED 302, such as authenticating received messages, and generating hash digests or authentication tokens. The CPU 304 also uses logic included in the memory 306 to perform the functions of the TED 302. When the CPU 304 generates an authentication token, it employs a secure one-way hash function on some content, such as a message, to generate a hash digest of that content. A one-way secure hash is a cryptographic operation where an input is run through some mathematical operations to produce an output, the hash digest, which is of fixed-length and which is probably unique. The hash digest has at least two properties: (1) determining the input to the hash function, given the hash digest, is virtually impossible or is at least computationally difficult; and (2) a hash digest for an input is essentially unique. The probability that two different inputs will result in the same output is extremely small. All of the hash digests discussed in this disclosure are generated from secure one-way hash functions.

The TED 302 uses EMMs to communicate entitlements and MSKs for programs or instances of service to the DSCTs 110. The TED 302 receives an EMM template from the EMM generator 320 addressed to a specific DSCT 110. The CPU 304 completes the template, generates a hash digest of the message content and includes the hash digest as an authentication token in the EMM. The CPU 304 retrieves the public key of the specific DSCT 110 from the CAA/TED database 322, and the content of the EMM is encrypted by the cryptographic accelerator 308 using that public key. Typically, the authentication token is signed by the cryptographic accelerator 308 using the private key of the TED 302. The EMM is signed so that the specific DSCT 110 can confirm that the EMM did in fact come from the TED 302.

The TED 302 can also receive messages from the DSCT 110. If a message was signed by a given DSCT 110, the CPU 304 gets the public key of the given DSCT 110 from the CAA/TED database 322, and the cryptographic accelerator 308 authenticates that the message did in fact come from the DSCT 110. If the message includes content

that was encrypted by the public key of the TED 302, then the cryptographic accelerator 308 uses the private key of the TED 302 to decrypt the content of the message. Frequently, the portion of the message that was signed is a hash digest of the message. In that case, the CPU 304 can generate a hash digest of the decrypted content of the message and compare the generated hash digest with the received hash digest. If both digests are the same, then the CPU 304 determines that the message was not corrupted in transmission nor tampered with.

The CAA 312 is the trusted cryptographic authority in the DBDS 100, which means that any message that is signed by the private key of the CAA 312 is to be considered by the recipient of that message as a valid or authentic message. As the trusted authority, the CAA 312, among other things, certifies the public keys of the TEDs 302. The CAA 312 includes a central processing unit (CPU) 314, a memory 316 and a cryptographic accelerator 318. The memory 316 includes three public key-private key pairs for the CAA 312, which are used by the CPU 314 for digitally signing EMMs and establishing and disestablishing TEDs 302 with DSCTs 110. The EMMs are signed so that the recipient of an EMM knows that the EMM came from the CAA 312. Each of the private keys can be also used for decrypting messages that have been encrypted by the corresponding public key of the CAA 312.

The DSCTs 110 are provided with the public keys of the CAA 312 so that they can authenticate EMMs coming from the CAA 312. In the preferred embodiment, the public key of the CAA 312 is included in the DSCT 110 as part of the manufacturing process of the DSCT 110. In an alternative embodiment, the public key of the CAA 312 is provided to the DSCTs 110 after the DSCT 110 has been manufactured and prior to the DSCT 110 being installed in the DBDS 100.

The memory 316 also includes the logic for performing the functions of the CAA 312, such as authenticating received messages and generating hash digests or authentication tokens. The CPU 314 uses logic included in the memory 316 to perform the functions of the CAA 312.

The CAA 312 communicates commands and information to the DSCTs 110 through EMMs. The CAA 312 receives an EMM template from the EMM generator 320 for a specific DSCT 110. The CPU 314 completes the template according to the type of command or information that it wants to convey. For example, the CPU 314 can command a DSCT 110 to establish a TED 302 or disestablish an established TED. Then the CPU 314 generates a hash digest of the message content and includes the digest as an

authentication token in the EMM. Typically, the authentication token is signed by the cryptographic accelerator 318 using the private key of the CAA 312 so that the specific DSCT 110 can confirm that the EMM did in fact come from the CAA 312. When it is necessary for the content of the EMM to be protected, the CPU 314 gets the public key of the specific DSCT 110 from the CAA/TED database 322, and, in that case, the content of the EMM is encrypted by the cryptographic accelerator 318 using the public key of the DSCT 110.

To establish a TED 302 with a DSCT 110, the CAA 312 sends an EMM to a particular DSCT 110 with instructions for allocating a portion of its memory to a given TED 302. Because the DSCT 110 already has the public key of the CAA 312, the DSCT 110 can authenticate the EMM as having come from the CAA 312. The CAA 312 provides the DSCT 110 with the public key of the TED 302 in an EMM, thereby establishing the TED 302 in the DSCT 110. Now that the DSCT 110 has the public key of the TED 302, the DSCT 110 and the TED 302 can securely communicate. The CAA 312 can also disestablish the TED 302, by sending the DSCT 110 an EMM telling the DSCT 110 that the DSCT 110 should no longer allocate any of its memory to the TED 302. For details of allocating and configuring memory in the DSCTs see U.S. Patent No. 5,742,677, Pinder, et al., *Information Terminal Having Reconfigurable Memory,* filed 4/3/95, which is hereby incorporated by reference in its entirety.

The CAA 312 can also receive messages from the DSCT 110. If the message was signed by the DSCT 110, the CPU 314 gets the public key of the DSCT 110 from the CAA/TED database 322 and the cryptographic accelerator 318 authenticates that the message did in fact come from the DSCT 110. If the message included content that was encrypted by the public key of the CAA 312, then the cryptographic accelerator 318 uses the private key of the CAA 312 to decrypt the content of the message.

The CAA 312 also grants limited authority within the DBDS 100 to the TED 302 and controls the types of services that the TED 302 can provide the DSCT 110. In a non-limiting example, the CAA 312 can determine that the TED 302 is entitled to provide interactive television services but not Internet services. Among other methods, the CAA 312 controls the types of services that the TED 302 can provide by the allocation of memory in the DSCT 110.

Transport Stream

The programs and instance of services provided by the entitlement agents are included in transport streams 240 and 242, which include packets of information. An instance of service includes, but is not limited to, a video service instance, an audio service instance, and a television program instance such as one episode of the evening news, among others. The preferred embodiment of the invention shall describe the packets of information as MPEG packets, but this is for illustrative purposes only and is a non-limiting example. The present invention is not limited to application to MPEG packets.

Referring to FIG. 4, for the sake of clarity a brief description of network transport stream 242 is provided hereinbelow. Network transport stream 242, which is a representative MPEG transport stream, is made up of a plurality of MPEG packets 400. Each of the MPEG packets 400 has a header 402 and a payload 404. The header 402 includes a packet identifier (PID) 406 that is used to identify the packet, such as 0, 258, 500, 1, etc. (PID values range from 0 to 8,191.) Certain packets, such as program association tables (PATs), which are identified by the PID value of 0, have reserved PID values. PATs are used to associate programs with program map tables (PMTs), which are used to identify the PID values of the elementary streams of the programs. For example, the exemplary PAT shown in FIG. 4, associates a program number 16 with a PMT packet having a PID value of 256, program number 25 with a PMT having a PID value of 500, etc. Generally, a program is made up of a plurality of elementary streams, and each one of the elementary streams in transport stream 242 has a unique PID value. The exemplary PMT, shown in FIG. 4, for program number 16 lists the elementary streams of program 16 and their respective PID values. For example, the video packets of program 16 are in the packets identified by the PID value of 257, and the PID stream 258 is the first audio stream of program 16.

System Encryption Scheme

Referring to FIG. 5, the encryption scheme implemented in the DBDS 100 is a multi-tiered scheme. In the preferred embodiment, CPU 304 and memory 306 (FIG. 5) of the TED 302 includes the logic used to perform the functions discussed below related to generating long-term keys or multi-session keys (MSKs).

A first tier has the highest tier of security in the DBDS 100. This tier is used for, without limitation, establishing the TED 302 of an entitlement agent with a DSCT 110,

creating the entitlements of the TED 302 in the DSCT 110, providing the DSCT 110 with the entitlements to access programs and instances of services, and providing the DSCT 110 with multi-session keys (MSKs) used for accessing the entitled instances of service and programs.

A second tier is related to the generation and transmission of relatively short term keys, or control words, control words being keys that are changed less frequently than the MSK. The control words are not as well protected as the MSKs, but they are changed so frequently that the security of the DBDS 100 is not seriously comprised if a control word is stolen. Generally, the control words are changed every couple of seconds, in one implementation, among others. In that case, a single stolen key will decrypt at most only a couple of seconds of unauthorized access.

A third tier is the lowest tier of encryption. The content of transport stream 240 is encrypted using a symmetrical cryptographic algorithm such as DES, 3DES, DVB common scrambling, or other encryption schemes known to those skilled in the art. Symmetrical cryptographic algorithms are chosen for the speed by which the encryption and decryption can be accomplished at the headend 102 and at the DSCT 110.

At the headend 102, the first tier includes portions of the system controller 232, including the EMM generator 320, the CAA/TED database 322, the CAA 312 and the TED 302. The EMM generator 320 creates EMM templates for both the TEDs 302 and the CAA 312 because both the CAA 312 and the TED 302 can send EMMs. Sometimes the content of the EMM does not need to be encrypted such as when the content is a public key. (Public keys are given out to the public, so there is no reason to protect them by encryption.) However, EMMs are frequently employed to transmit private information, and in that case, when privacy is desired, the content of the EMM encrypted with the public key of recipient. The CAA 312 and the TED 302 each include logic for determining when to encrypt or not encrypt a message content. The CAA 312 and the TEDs 302 handle EMMs in the same manner, so a description of how the CAA 312 handles EMMs is not provided.

In the preferred embodiment, the EMM messages are the most secure messages transmitted through the DBDS 100. The EMMs are used by the TED 302 to provide, among other things, each DSCT 110, for which the TED 302 is established, with decryption information, such as MSKs, for programs or instances of service and authorization that are used for the DSCT 110 to access a program or instance of service that is associated with the TED 302. For example, if a subscriber decides to order a pay-

per-view movie, the subscriber receives an EMM that includes the authorizations for the DSCT 110 of the subscriber to access the ordered pay-per-view movie. EMMs are also used, for among other things, for distributing MSKs. Generally, MSKs have an expiration date, and as such new MSKs need to be sent before the expiration of the current MSK. Each DSCT 110 that is associated with the TED 302 receives an EMM having a new MSK prior to the expiration of the current MSK.

The EMM generator 320 includes logic for checking the CAA/TED database 322 and determining the expiration date of the current MSK for the TED 302, and then addressing the EMM template for each DSCT 110 that is associated with the TED 302.

The TED 302 receives an EMM template, which is addressed to one of the DSCTs 110 associated with the TED 302, and the CPU 302 implements logic stored in the memory 306 for implementing a multi-session key generator (not shown), which is used for generating multi-session keys (MSKs) 522.

The CPU 304 creates an MSK 522, which is then used as part of the encryption/decryption scheme implemented in DBDS 100, and copies of the MSK 522 are sent to the CAA/TED database 322, along with its expiration date, and to the multi-transport stream receiver-transmitter 228. In an alternative embodiment, a copy of the MSK 522 is sent to the CAA/TED database 322, and the multi-transport stream receiver-transmitter 228 retrieves the MSK 522 from the CAA/TED database 322 when it needs it.

The CPU 304 employs a one-way hash function using a portion or all of the MSK to produce a hash digest. The hash is used as an authentication token. The DSCT 110 that receives the MSK in an EMM can determine whether the EMM was corrupted in transmission by creating another hash digest of the EMM and comparing its hash digest with the received authentication token. If the two are the same, then the EMM was not corrupted. In the preferred embodiment, the cryptographic accelerator 308 uses the private key of the TED 302 to sign the authentication token. The content of the message is then encrypted by the CPU 304 using the public key of the DSCT 110, and the encrypted content, or ciphertext, and the signed authentication token are included in the EMM, which is sent to the DSCT 110. Thus, the EMM having the MSK has two levels of security: the encrypted content and the signed authentication token.

Each time the MSK generator 504 produces a new MSK 522, the MSK 522 is sent to the control word generator 510 and to the per-TED database of the CAA/TED database 322 along with its expiration. The MSK 522 is changed as frequently as the operators of DBDS 100 desire to do so.

Each DSCT 110 of the DBDS 100 has its own public key-private key pair, and the EMM's content is frequently encrypted using the public key of the DSCT 110 to which the EMM is addressed. (It is not necessary to encrypt the content of all of the EMMs. Some EMMs are used to transmit copies of public keys that don't need to be kept secret. However, these EMMs include an authentication token is used to protect the content of these messages.) Consequently, only the particular DSCT 110 that an EMM is addressed to has the correct private key for decrypting the EMM, in accordance with this embodiment, among others. The particular DSCT 110 decrypts the encrypted content using its private key and validates the signature of the TED 302 applied to the authentication token using the public key of the TED 302. The DSCT 110 then creates a hash digest of at least a portion of the contents of the EMM and compares the hash digest with the received authentication token. Provided the produced hash digest and the received authentication token are the same, the DSCT 110 then knows that the EMM has not been tampered with or corrupted in transmission. If the content of the EMM was altered, then the hash digest and the authentication token will not be the same, and in that case, the EMM is ignored. EMMs addressed to the DSCT 110 are repeatedly sent, and therefore, if an EMM is corrupted in transmission, an uncorrupted EMM should arrive at the DSCT shortly.

At the headend 102, the second and third tiers are preferably implemented in the system controller 232 and the multi-transport stream receiver-transmitter 228. The system controller 232 includes an entitlement control message (ECM) generator 512, which is used for generating ECM templates, and the ECM templates include information that associates the ECM template with a particular TED 302 and with a program or instance of service provided by the entitlement agent associated with the particular TED 302. The ECM template also includes information about the entitlements that are needed by the DSCTs 110 to access the associated program or instance of service. A DSCT 110 that receives the ECM for an associated program or instance of service checks its authorizations for programs or instances of service, and only if the DSCT 110 has the proper authorization will the DSCT 110 use the ECM to access the program or instance of service.

Due to efficiency concerns, symmetric cryptographic algorithms are employed for encrypting the packets 400 of the transport stream 240 and the content of the ECMs. Those skilled in the art recognize that symmetric cryptographic algorithms are generally faster than asymmetrical cryptographic algorithms, but symmetric cryptographic

algorithms can also be less secure, because a mechanism is required to provide the same key to both the encryptor and decryptor. However, the security of the DBDS 100 is not at risk. As previously stated hereinabove, the control word 524, which is used as the encryption key, is frequently changed. Preferably it is changed every couple of seconds. Therefore, gaining access to a particular instance of service requires knowledge of multiple control words; more than 1,000 control words are used for an instance of service that lasts for one hour in which the control word is replaced every 3.5 seconds. Symmetric cryptographic algorithm include, but are not limited to, DES, 3DES and DVB Common Scrambling.

The multi-transport stream transmitter receiver 228 includes, at least, a control word generator 510, a multiplexer 516, a cryptographic device 518, and a modulator 520. The control word generator 510 produces a control word 524, which is provided to the cryptographic device 518 as a key to be used with a function of a cryptographic algorithm. The control word 524 can be either a number produced by a random number generator (not shown) or a "pseudo-random" number. In the preferred embodiment, the count of a sequential counter (not shown) is encrypted to produce a pseudo-random number or the control word 524. The control word generator 510 receives the MSK 522 from the TED 302 and uses the MSK to encrypt the counter. In the preferred embodiment, the counter is encrypted using a cryptographic algorithm such as 3DES, or other symmetrical cryptographic algorithms.

The control word generator 510 produces ECMs that are sent to the DSCTs 110 using an ECM template received from the ECM generator 512. The ECMs include the counter, which is transmitted in the clear, and an authentication token produced by the control word generator 510. The authentication token is a one-way hash digest of at least a portion of the ECM, such as the counter, and a secret that is shared with the DSCTs 110. The secret is the MSK 522, which the established DSCTs have already received via an EMM. In the preferred embodiment, the control word generator 510 creates a new control word 524 every few seconds. Each time a new control word 524 is produced, the control word generator 510 provides the cryptographic device 518 with the new control word 524 and includes the new counter in a new ECM, which is then sent to the DSCTs 110. In an alternative embodiment, the authentication token is a one-way hash digest of at least a portion of the MSK 522, the control word 524 and the message content of the ECM.

In an alternative embodiment, a true random number is used as the control word 524. In that case, the random number is encrypted by a symmetrical cryptographic algorithm using the MSK as a key and the encrypted control word is included in the ECM. A hash digest using at least the control word 524 is produced by the control word generator 510 and included in the ECM. The control word 524 is provided to the cryptographic device 518, and the ECM is sent to the DSCTs 110. The random number generator produces a new control word 524 every couple of seconds; and the new control word 524 is provided to the cryptographic device 518, and a new ECM having the new control word 524 included therein is created and sent to the DSCTs 110.

The cryptographic device 518 receives the control word 524 and uses it to encrypt the packets of transport stream 240. In the preferred embodiment, the cryptographic device 518 is adapted to perform both encryption and decryption functions of a cryptographic algorithm such as DES, or 3DES, or DVB common scrambling, and other cryptographic algorithms known to those skilled in the art. Thus, when input transport stream 240 includes encrypted content, the cryptographic device 518 can apply a function of a cryptographic algorithm to further encrypt the content, i.e., add another layer or encryption to the content, or apply a function of a cryptographic algorithm to remove a layer of encryption. Typically, the control system 232 provides the multi-transport stream transmitter-receiver 228 with the keys that are used by the cryptographic device 518 to remove a layer of encryption. In the preferred embodiment, some of the content of the input transport streams 240 is encrypted with a key different than the control word 524, and in that case, the control system 232 provides the multi-transport stream transmitter-receiver 228 with the encryption key.

Typically, the cryptographic device 518 receives the transport stream 240 and applies a function of the cryptographic algorithm with the control word to the content of transport stream 240. The cryptographic device 518 applies a function of a cryptographic algorithm to the payload 404 of the packet 400 using the control word 524 as the key, thereby converting the packets 400 of transport stream 240 into encrypted packets 528.

The multi-transport stream transmitter receiver 228 includes a multiplexer 516 and a modulator 520. In one embodiment, the multiplexer 516 receives EMMs from the TED 302, ECMs from the control word generator 510, and encrypted packets 528 and multiplexes them into transport stream 508. The modulator 520 receives the transport stream 508 and outputs modulated transport stream 242, which is then received by the DSCTs 110.

Subscriber Location 108

Referring back to FIG. 1, to prevent unauthorized access to (or copying of) programs or instances of service transmitted to the DSCTs 110, the programs or instances of service are encrypted at the DSCT 110 prior to storing the program or instance of service at the subscriber's location 108. In the preferred embodiment, the key or keys used for encrypting the program or instance of service are then restricted such that the keys work only with the DSCT 110 that recorded the program or instance of service. An alternate method for restricting access to the key(s) used for decrypting programs or instances of service is for the headend or the content provider 114 to control the key(s).

DSCT 110

The DSCT 110 receives programming or instances of service from the headend 102. The DSCT 110 is adapted to respond to user commands to process the received programs or instances of service such that programs or instances of service can be provided to a user device such as, but not limited to, the television 112. The DSCT 110 is also adapted to process programs or instances of service and store the programs or instances of service at the subscriber's location 108. Three examples for encrypting and storing the content of a program or instance of service at the subscriber's location are described hereinbelow. These examples are non-limiting examples for illustrative purposes only, and those skilled in the art will recognize other embodiments, which are intended to be included within the scope of the invention. However, first a description of an example DSCT 110, among others, and a description how the DSCT 110 receives and processes a program or instance of service are provided.

Referring to FIG. 6, the DSCT 110 is coupled to communication link 154, which includes in-band communication 254 and out-of-band communication 256 (FIG. 2). The DSCT 110 includes a tuner 602, a transceiver 604, a demultiplexer 606, a processor 608, a cryptographic device 610, a secure processor 612, a storage device 614, an input/output interface 616, a converter 618, and a memory 626, among other elements.

The tuner 602 responds to subscriber commands issued via a user interface device such as a remote control (not shown). The commands are received by an infra-red receiver (not shown), which sends the commands to the processor 608. The processor 608 and memory 626 include the logic for implementing commands from the user interface device (not shown).

A subscriber typically uses the remote control to select a "channel" (from a conventional user perspective) that is associated at a given time with an instance of service or program provided by an entitlement agent. The processor 608 uses tables such as network information tables (NITs) stored in memory 626 to determine the frequency band or other selection criteria associated with the user selected "channel." The processor 608 then instructs the tuner to tune to that particular frequency band. The instructions are relayed from the processor 608 to the tuner 602 via bus 620. Other methods of interacting with the user and selecting a particular instance of service could be used.

The tuner 602 provides the demultiplexer 606 with the transport stream 242 that is contained in the frequency band to which the tuner is tuned. The demultiplexer 606 extracts system tables such as NITs, PATs and CATs, which are included in packets having reserved PID values and provides the tables to the processor 608 via bus 620. Typically, the system tables are periodically included in a transport stream, and when a new system table is extracted by the demultiplexer 606, the new table is sent to the processor 608. Typically, some of the system tables are stored in memory 626, so that the processor 608 can have prompt access to them.

In addition to extracting system tables and specific packets determined by the processor 608, the demultiplexer 606 extracts ECM packets, which are then provided to the secure processor 612, via bus 620, for processing. In one embodiment the EMMs are transmitted to the DSCT 110 in the in-band communication path 254, and the demultiplexer 606 extracts the EMMs and provides them to the secure processor 612.

Typically, a transport stream in a frequency band includes multiple multiplexed programs or instances of service, and each program or instance of service is associated at a given time with a user "channel." Thus, in response to a user selecting a new "channel," the processor 608 uses tables such as PATs and PMTs to determine the PID values of the elementary streams that make up the instance of service or program associated with the selected user "channel" at that time. It should be remembered that the new "channel" is part of a transport stream and that the new channel may be included in the frequency band of the old channel. It is to be understood that a user channel represents one type of communication channel. Communication channels include, but are not limited to, communication signals that are separated by: frequency, which is generally referred to as frequency-division multiplexing (FDM); time, which is generally referred to as time-division multiplexing (TDM); and code, which is generally referred to as code-division multiplexing (CDM), among others. In the preferred embodiment, a frequency

band having a transport stream included therein is 6 Mhz wide, and the in-band communication path 254 includes multiple 6 Mhz bands. Historically, one analog television signal was broadcast in a 6 Mhz band, and consequently, each band was considered a television channel. This historical approach is no longer accurate because with the advent of multiplexing each of the 6 Mhz bandwidths can contain multiple communication signals. Thus, it is to be understood that a channel does not necessarily specify a frequency band, rather it simply refers to a communication signal in some contexts.

The processor 608 sends the PID values of the elementary streams to the demultiplexer 606 via bus 620, and the demultiplexer 606 extracts the packets having the PID values identified by the processor 608. The demultiplexer 606 processes the elementary streams according to instructions received from the processor 608 and sends the packets of the elementary streams to either cryptographic device 610, which is a cryptographic device that encrypts and decrypts information, for further processing or to a storage device. After the cryptographic device 610 has processed the packets, the packets are then sent to either storage device 614 or to the input/output interface 616 for storage in an external storage device 650 or to another external device (not shown). Non-limiting examples of storage devices include, but are not limited to, hard-drives, CDs, magnetic tape, DVDs, and media servers.

The transceiver 604 is used by processor 608 for two-way communication via the out-of-band communication path 256 with the headend 102 or HUB 104. Frequently, the communications from the processor 608 to the headend 102 or HUB 104 include requests for services such as receiving a video-on-demand program or instance of service. In addition, the communications from the headend 102 or hub 104 can include commands. For example, the processor 608 can receive from the headend 102 instructions for storing a program or instance of service. In the preferred embodiment, the commands include instructions for tuning to a particular frequency band at a predetermined time, extracting a specific program or instance of service included in the transport stream that is modulated at the particular frequency band and processing the program or instance of service of that transport stream. If the DSCT 110 is inactive, i.e., not being used by the subscriber, the processor 608 responds to those instructions by having the tuner 602 tune to the appropriate frequency at the predetermined time. The processor 608 has the demultiplexer 606 extract from the transport stream 242 the elementary streams for the specific program or instance of service. The demultiplexer 606 processes the elementary

streams according to instructions relayed from the processor 606 and sends the packets of the elementary streams to either cryptographic device 610 for further processing or to a storage device.

The processor 608 sends the cryptographic device 610 instructions regarding how the cryptographic device 610 should process elementary stream provided by the demultiplexer 606 and where the cryptographic device 610 should send the processed elementary streams. In the preferred embodiment, the cryptographic device 610 is adapted to perform multiple functions of a cryptographic algorithm on the payload portion 404 of the packets 400 (FIG. 4) that make up the received elementary streams, and it is adapted to perform more than one type of cryptographic algorithm.

Typically, when the cryptographic device 610 receives packets 400 (FIG. 4) which are encrypted, from the demultiplexer, the cryptographic device 610 obtains the control word 524 from the secure element 612 and decrypts the packets. The packets 400 are then sent to the converter 618 so that they can be converted to the appropriate format for a user device such as, but not limited to, a TV 112, VCR, or computer.

Sometimes the processor 608 tells the cryptographic device 610 that the packets 400 are to be sent to a storage device such as storage device 614, or storage device 650 via the input/output interface 616. In that case, the cryptographic device 610 can process the packets in at least several different ways. In one non-limiting case, it can get instructions from the processor 608 to decrypt the packets using the control word 524 and then re-encrypt the payload 404 of the packets 400 using an encryption key, or media key. The media key, which is used as a key for encrypting and decrypting content stored at the subscriber's location, can be generated by the processor 608, the secure element 612, or the system controller 232. The re-encrypted packets are then sent to the storage device 614, or to the input/output interface 616 along with the media key, which is associated with the packets. The media key can be used to encrypt all of the packets that make up a program or instance of service, or multiple media keys can be used to encrypt different packets of the program or instance of service. In one embodiment, the packets that make up a program or instance of service are encrypted by multiple media keys including the control word 524.

In another non-limiting case, the cryptographic device 610 receives packets having encrypted content and gets instructions from the processor 608 to further encrypt the payload 404. In that case, the cryptographic device 610 gets a media key and uses it to convert the ciphertext of the payload 404 to a different ciphertext 404. The

cryptographic device 610 then sends the processed packets along with the media key to the storage device 614, or to the input/output interface 616. The media key is then associated with the packets and sent along with the packets for storage. Again, a single media key can be used to encrypt all or some of the packets that make up a program or instance of service.

In another non-limiting case, the encrypted packets received from the demultiplexer 606 can be processed by the cryptographic device 610 multiple times using multiple keys and multiple functions of a cryptographic algorithm according to instructions from the processor 608. The cryptographic device 610 then sends the processed packets and at least one of the multiple keys for storage to the storage device 614 or to the input/output interface 616.

In yet another non-limiting case, the cryptographic device 610 receives packets from the demultiplexer 606 that are not encrypted, i.e., clear text packets, the cryptographic device 610 receives a media key and uses the media key with a function of a cryptographic algorithm to encrypt the packets. Again, the media key can be generated at the headend, or by the processor 608, or by the secure element 612. The encrypted packets and the media key are sent to either the storage device 614 or to the input/output interface 616 for storage.

Thus, in the preferred embodiment, the cryptographic device 610 can do at least one or more of the following: (1) receive cleartext packets, encrypt them, and send them to a storage device; (2) receive encrypted packets, decrypt them, and send them to a storage device, or to the converter 618, or re-encrypt them and then send them to a storage device; and (3) receive encrypted packets, further encrypt them, and send them to a storage device.

Downloadable programs and services can also be received via out-of-band communication path 256. The packets of the downloaded program or instance of service can be sent from the transceiver 604 directly to a storage device such as storage device 614 or 650, and/or they can be sent to the cryptographic device 610 for processing prior to storage.

The subscriber can use his or her DSCT 110 user interface, such as a remote control, to retrieve stored programs or instances of service. The processor 608 responds to user commands and checks whether the stored program or instance of service is encrypted. If it is not encrypted the user is provided with the program or instance of service via the converter 618.

If the stored program or instance of service is encrypted, the processor 608 provides the cryptographic device 610 with the media key, or media keys, and the corresponding packets of the program or instances of service. The cryptographic device 610 decrypts the program or instance of service and sends the decrypted program or instance of service to the user device via converter 618. In the preferred embodiment, the processor 608 first determines whether the user has permission to access the program or instance of service before the program or instance of service is provided to the cryptographic device 610; and if the subscriber does not have permission, the media key and the program and the instance of service are not provided to the cryptographic device 610.

The secure processor 612 includes a processor 622 and a memory 624. In the preferred embodiment, the secure processor 612 is in tamper proof packaging to prevent unauthorized persons from accessing processor 622 and memory 624. The memory 624 is accessible only to the processor 622. The secure processor 612 includes the logic for processing EMMs and ECMs and for providing control words 524 to the cryptographic device 610 via bus 620 for decrypting received encrypted programs or instances of service.

The logic of the secure processor 612 also enables the processor 622 to configure and allocate a portion of the memory 624 to a TED 302 to establish the TED 302 with the DSCT. The processor 622 configures and allocates the memory 624 in response to EMMs from the CAA 312. The memory 624 includes the private key of the DSCT 110 and the public key of each TED 302 that is established with a DSCT 110. The private key of the DSCT 110 is kept within the secure element 612 and is not accessible to components outside of the secure element 612. The secure element 612 also includes the logic for generating encryption keys that are used for encrypting programs or instances of services or other content that is stored in the storage device 614 or storage device 650.

In order to access a program or instance of service associated with a particular TED 302, the DSCT 110 needs the MSK 522 provided by the TED 302 and the control word 524 that was used for encrypting packets of the program or instance of service. The DSCT 110 is provided with the MSK 522 and an EMM from the TED 302 and with indicators used for producing the control word 524 in the ECMs. As described hereinbelow, the secure element 612 processes the EMMs and ECMs to generate the control word 524 and determine whether the DSCT 110 is entitled to the program or instance of service.

Referring to FIG. 7, which shows in the preferred embodiment, selected elements of the DSCT 110, among many other elements, the secure element 612 includes a message decryptor 702, a message authenticator 704, a service authorizer 706, a control word generator 708, and a key repository 710, which are embodied in the logic of the processor 622 and the memory 624.

The demultiplexer 606 receives transport stream 242 and sends the EMMs and ECMs that are included in transport stream 242 to the secure processor 612 for processing. The demultiplexer 606 also sends the encrypted elementary streams of selected programs or instances of service to the cryptographic device 610 for decryption.

The EMM having the MSK 522 that was used in the generation and/or protection of the control word 524 is received at the DSCT 110 prior to receiving the content that was encrypted using the control word 524.

The key repository 710 has the public key-private key pair of keys for the DSCT 110 stored therein and the public keys that the DSCT 110 has received. Received public keys include the public keys of established TEDs 302. The key repository 710 is also used for storing MSKs 522.

The message decryptor 702 receives an EMM and retrieves the private key of the DSCT 110 from the key repository 710 and uses the private key to decrypt the content of the EMM. The cleartext or decrypted message content and the authentication token of the EMM are provided to the message authenticator 704. The message authenticator 704 validates the authentication token of the EMM. To validate the authentication token of the EMM, the message authenticator 704 checks to see if the purported sender of the EMM did actually send the message and then checks the content of the message. First, the message authenticator 704 obtains from the key repository 710 the public key that is associated with the TED 302 that purportedly sent the EMM and uses the public key to process the digital signature applied to the authentication token to recover a value. If the public key corresponds to the private key that applied the digital signature to the authentication token, then the recovered value represents the hash digest of the message. Otherwise some other value is recovered. Next, the message authenticator creates a hash digest of at least a portion of the decrypted content of the EMM. The hash digest is compared with the recovered value, and if they are the same, the EMM is valid. If the contents of the EMM had been altered by the subscriber or other unauthorized persons or corrupted in transmission, the hash digest and recovered value would not be the same. In that case, the EMM would be ignored in this embodiment. The verification also fails if

the public key used to recover the value was not the corresponding key because hash digest of the message content will not match the recovered value.

As previously described hereinabove, the contents of the EMM are generally service authorizations or keys. When the content of the EMM includes service authorizations, i.e., authorizations for instances of service and programs provided by the entitlement agent associated with the TED 302 to the DSCT 110, the service authorizations are provided to the service authorizer 706. When the EMM content is a key, such as an MSK 522, the key is stored in the key repository 710. However, it should be noted that the contents of the EMM are preferably only acted upon provided: (1) the EMM was addressed to the DSCT 110; (2) the EMM was actually signed by the TED that purportedly sent the EMM; and (3) the contents of the EMM have not been altered or corrupted. Of course, other embodiments may not require all of these security features. In the preferred embodiment of the invention, the EMMs are processed in the first tier of security.

Before the program or instance of service can be decrypted by the cryptographic device 610, the ECM, which is in the second tier of security, must be received and processed by the secure processor 612. In the preferred embodiment, the ECM includes the cleartext counter that is used to generate the control word 524 and an authentication token. The message authenticator 704 retrieves the MSK 522 from the key repository 710 and creates a hash digest of at least a portion of the MSK 522 and at least a portion of the content of the ECM. The hash digest is compared to the authentication token, and if they are the same, the ECM is regarded as valid. In that case, the counter is sent to the control word generator 708.

The service authorizer 706 includes the authorizations for services provided to the DSCT 110 by the entitlement agent. The service authorizer 706 checks the ECM and determines whether the DSCT 110 is authorized for a particular program or instance of service provided by the entitlement agent. When the DSCT 110 is authorized, the service authorizer 706 provides the control word generator 708 with the MSK 522.

The control word generator 708 receives the control word, which is a number, from the message authenticator 704 and encrypts the control word using the MSK 522 to produce the control word 524. The control word generator 708 provides the control word 524 to the cryptographic device 610, and finally, the cryptographic device 610 uses the control word 524 to decrypt the elementary streams of the selected program or instance of service.

In another embodiment, the ECM includes the cleartext counter and an authentication token, which is the digest of a one-way hash function having as inputs at least a portion of the control word 524, the message content of the ECM and the MSK 522. In this case, the message authenticator 704 retrieves the MSK 522 from the key repository 710 and provides the MSK 522 and the cleartext counter to the control word generator 708. The control word generator 708 generates the control word 524 and returns it to the message authenticator 704, which uses it, the MSK 524 and the content of the ECM to generate a hash digest. The hash digest generated by the message authenticator 704 is compared with the authentication token included with the ECM and if they are the same, the message authenticator 704 validates the message as being authentic. In response to the message authenticator 704 validating the ECM, the service authorizer 706 receives the control word 524 from the message authenticator 704 and provides the control word 524 to the cryptographic device 610.

In yet another embodiment, the ECM includes an encrypted control word 524 that was encrypted by a cryptographic algorithm using the MSK 522 as a key. In that case, the message decryptor 702 retrieves the MSK 522 from the key repository 710 and uses the MSK 522 to decrypt the contents of the ECM. The control word 524 is then provided to the service authorizer 706, which will provide the cryptographic device 610 with the control word 524 only if the DSCT 110 is authorized for the program or instance of service associated with the control word 524. The encrypted elementary streams of the programs of instances of service are in the third tier of security.

The multi-tiered encryption scheme offers a number of advantages with regard to security. It takes advantage of the speed of symmetrical encryption system where speed is useful to encrypt and decrypt the payload 404 of packets 400 and to produce the control word 524. The control word 524 is protected in the ECM by encrypting it using the MSK 522 and by including an authentication token in the ECM. The authentication token includes a portion or all of the MSK 522as a shared secret between the TED 302 and the DSCT 110. The MSK 522 is protected in turn by the fact that it is sent in an EMM that is encrypted using the DSCT's public key and by the fact that the EMM includes a sealed digest that is signed by the entitlement agents private key. Further, security is provided by the fact that service identification information from the ECM must agree with authorization information received in an EMM before the control word 522 is provided to the service decryptor 610.

Referring to FIG. 8, in accordance with the preferred embodiment, steps 800 are performed at the subscriber's DSCT 110 to process received ciphertext, when the ciphertext is to be stored at the subscriber location 108. In step 802, the DSCT 110 receives a ciphertext packet from the headend 102. The ciphertext packet includes the payload 404, which is encrypted. The encrypted payload, or ciphertext, can have one or more layers of encryption, wherein a layer of encryption corresponds to applying a function of a cryptographic algorithm to the payload portion of the packet. For example, for the purposes of this disclosure, a payload that has been twice encrypted using the DES cryptographic algorithm is considered to have 2 layers of encryption. Generally, the ciphertext packet conforms to a packet such as an MPEG packet, but the present invention is not limited in scope to MPEG packets or packets in general.

The headend 102 sends the ciphertext packet and a key to the DSCT 110 via transport stream 254, but, in alternative embodiment, the ciphertext packet and the key are sent to the DSCT 110 via out-of-band communication 256. The key is a decryption key for the ciphertext packet. In one embodiment, the decryption key is a control word 524, which is included in an ECM. However, in an alternative embodiment, among others, the ciphertext packet was encrypted at the headend by a key, which has a lifetime that is longer than a control word 524, such as MSK 522. In that case, the DSCT 110 receives the MSK 522 via an EMM. In yet another embodiment, the key is retained at the headend 102 until the headend 102 receives a request from the subscriber for the key, or until the headend 102 determines that it is time to release the key.

In step 804, the cryptographic device 610 applies a cryptographic algorithm at least once to the received ciphertext packet to produce a different, or new, ciphertext packet. In the preferred embodiment, the cryptographic algorithm uses at least one key produced at the DSCT 110 to encrypt or further encrypt the contents of the ciphertext packet. Typically, the received ciphertext is processed several times by the cryptographic algorithm using a plurality of keys to produce the new ciphertext packet. The processing includes applying multiple functions of a cryptographic algorithm. Generally, the plurality of keys includes keys that are generated at the DSCT 110 by the processor 608 or the secure element 610 and keys received at the DSCT 110.

In the preferred embodiment, the new ciphertext packet corresponds to a cleartext packet that was encrypted by a cryptographic algorithm different from the plurality of cryptographic algorithms used to produce it. For example, in the preferred embodiment, the cleartext packet was encrypted multiple times using the DES cryptographic algorithm

with a set of keys, thereby producing the new ciphertext packet. However, the new ciphertext packet is equivalent to a ciphertext packet that was produced by encrypting the cleartext packet using the 3DES cryptographic algorithm with the same set of keys. Those skilled in the art recognize that the DES cryptographic algorithm and the 3DES cryptographic algorithm are different cryptographic algorithms.

In step 806, the new ciphertext packet is stored at the subscriber location. The new ciphertext packet can be stored within the DSCT 110 or externally to the DSCT 110.

Three non-limiting examples of the implementation of steps 800 are provided hereinbelow. Those skilled in the art will recognize other examples and implementations that are intended to be within the scope of the invention.

EXAMPLE 1:

Referring to FIG. 9, which shows some of the functional components, among others, of the headend 102 and the DSCT 110, a packet of cleartext 902 is received at the headend 102. The cleartext packet 902 is part of a program or instance of service that is provided to the subscriber of DBDS 100 by the content provider 114, or entitlement agent. The cleartext packet 902 can be stored at the headend 102 in storage devices such as VOD pump 226. (FIG. 2)

At the headend 102, the cleartext packet 902 is provided to the cryptographic device 518 of the multi-transport stream receiver-transmitter 228 along with an encryption key 908. The cryptographic device 518 uses the encryption key 908 and the encryption function (E) of a cryptographic algorithm to convert the cleartext 902 to the ciphertext $C'\{E, k_1\}$ 906, which has one layer of encryption. In the preferred embodiment, the cleartext 902 packet and the ciphertext packet 906 are MPEG packets, and the payload portion 404 (FIG. 4) of the packet 400 is encrypted, and in the preferred embodiment, the cryptographic algorithm is the DES algorithm.

In an alternative embodiment, the cryptographic device 518 uses the encryption key 908 and the decrypt function (D) of a cryptographic algorithm to convert the cleartext packet 902 to the ciphertext packet 906, and the ciphertext packet 906 is then written as $C'\{D, k_1\}$ because it has one layer of encryption that was applied using the decryption (D) function with the key k_1. It should be remembered that the encryption function and the decryption function of a cryptographic algorithm are inverse functions. Either function, D or E, can be used to convert cleartext to ciphertext and then the other function, E or D, is then used to convert the ciphertext back to cleartext.

In an embodiment, the encryption key k_1 908 is a control word 524, which is frequently changed. Generally, the cryptographic device 518 uses a control word 524 such as $k_1(N)$ to encrypt multiple packets of a program or instance of service before the control word 524 is changed to $k_1(N+1)$. The ciphertext packet 906 is included in the transport stream 242 (FIG. 2) and transmitted to the subscriber's location 108. The key k_1 908 is included in ECMs or EMMs and sent to the DSCT 110. Typically, when multiple keys k_1 908 are used to encrypt a program or instance of service, the keys are included in ECMs. However, in the case when key k_1 908 is changed infrequently, or not at all during the transmission of the program or instance of service, the key(s) k_1 908 can be included in either an ECM(s) or EMM(s).

In an alternative embodiment, the headend 102 retains key k_1 908 and provides the key k_1 908 in response to subscriber requests. By withholding the key k_1 908 until receiving a request for the key from the subscriber, access to the content of the program or instance of service is controlled by the operators of the DBDS 100. This enables the operator of the DBDS 100 to download and store the programs or instances of service at the subscriber's location while preventing the subscriber from making useable bootleg copies and from accessing the program or instance of service until a predetermined time. The subscriber uses his or her user interface device (not shown) to request access to the stored program or instance of service, and the DSCT 110 then sends a request to the headend 102 for the key 908. The system controller 232 processes the request and sends the key 908 to the DSCT 110 and, if necessary, bills the subscriber for the program or instance of service.

At the subscriber location 108, the cryptographic device 610 receives the ciphertext packet 906 and uses a function of a cryptographic algorithm with the key k_2 912 to further encrypt the ciphertext packet 906. In the preferred embodiment, the decrypt function (D) of the DES cryptographic algorithm and the key k_2 912 are used to convert the ciphertext packet 906 to the ciphertext packet 914. The ciphertext packet 914, which is the result of adding a layer of encryption to the ciphertext packet 906, is written as $C''\{E, D; k_1, k_2\}$. The ciphertext packet 906 has two layers of encryption because it has been processed using an E function and a D function with keys k_1 and k_2, respectively. In alternative embodiments, the cryptographic algorithm is a symmetrical cryptographic algorithm known to those skilled in the art. In yet another embodiment, the function is the encrypt function (E) of a cryptographic algorithm such as DES.

The cryptographic device 610 reprocesses the ciphertext packet 914 using an encryption (E) function of a cryptographic algorithm and the key k_3 918 to further encrypt the ciphertext packet 914, thereby producing the ciphertext packet 920, which is written as $C'''\{E, D, E; k_1, k_2, k_3\}$. In the preferred embodiment, the cryptographic algorithm used for reprocessing the ciphertext packet 914 is the DES cryptographic algorithm. The ciphertext packet 920 corresponds to the cleartext packet 902 thrice encrypted by three different keys. In the preferred embodiment, cryptographic device 518 of the multi-transport stream receiver-transmitter 228 applies a first layer of encryption to the cleartext packet 902 using the encryption function (E) of the DES algorithm with the key k_1 908, and the cryptographic device 610 of the DSCT 110 applies a second and third layer of encryption using the decryption function (D) and the encryption (E) function of the DES algorithm with the keys k_2 912 and k_3 918, respectively, to produce the ciphertext packet 920. The ciphertext packet 920 corresponds to encrypting the cleartext packet 902 using the 3DES cryptographic algorithm with the keys k_1 908, k_2 912 and k_3 918.

The keys 912 and 918, which are used by the cryptographic device 610 for encrypting, are preferably generated locally in the DSCT 110 by the processor 608 or the secure processor 612, but, in an alternative embodiment, the keys 912 and 918 are generated at the headend 102 and sent to the DSCT 110 via EMMs or ECMs. In yet another embodiment, at least one of the keys 912 or 918 is generated at the headend 102 and provided to the DSCT 110 via EMMs or ECMs and the other key is generated locally at the DSCT 110.

In the preferred embodiment, the ciphertext packet 920 is stored in the storage device 614. Typically, storage device 614 is a hard-drive, CD, or other writeable electronic media known to those skilled in the art. In an alternative embodiment, the ciphertext packet 920 is stored in the external storage device 650 (FIG. 6).

In the preferred embodiment, the storage device 614 stores all of the ciphertext packets 920 of a program or instance of service 926 and the media keys 924, which are the keys used for decrypting the ciphertext packet 920 of the program or instance of service 926. The media keys 924 are then restricted to prevent the subscriber from providing unauthorized copies of the media keys 924 and the program or instance of service 926. When the subscriber wishes to use the stored program or instance of service 926, the subscriber interacts with the DSCT 110 using a subscriber interface device (not shown) such as a remote control and requests the stored program or instance of service 926. The ciphertext packets 920 of the stored program or instance of

service 926 are sent to the cryptographic device 610 along with the appropriate media keys 924, i.e., keys k_1 908, k_2 912, and k_3 918. The cryptographic device 610 uses the keys 908, 912, and 918 with the appropriate encryption and decryption functions of the cryptographic algorithm to convert the ciphertext 920 to cleartext 902.

As previously stated hereinabove, the key k_1 908 can also be stored at the headend 102. In that case, the key k_1 908 is then securely provided to the subscriber via EMMs or ECMs when the subscriber requests access to the stored program or instance of service, and it is provided to the cryptographic device 610.

In the preferred embodiment, the cryptographic device 518 and the cryptographic device 610 each perform at least one of the operations for converting cleartext packet 902 into a 3DES encrypted ciphertext packet 920, and the cryptographic device 610 converts the ciphertext packet 920 to the cleartext packet 902 using the decryption function of the 3DES cryptographic algorithm with the keys 908, 912 and 918.

In an alternative embodiment, the functions of the cryptographic device 610 are distributed in multiple cryptographic devices (not shown). In a non-limiting example, a first cryptographic device (not shown) processes the ciphertext packets 906 and 914 to produce the ciphertext packet 920, and a second cryptographic device (not shown) converts the ciphertext packet 920 to the cleartext packet 902.

Generally, the cleartext packet 902 is received by the converter 618, which converts the cleartext from a packetized format to a format appropriate for user device 112. Typically, the user device 112 is a television, a computer, a VCR or other such device.

EXAMPLE 2:

Referring to FIG. 10, a packet of cleartext 902 is received at the headend 102. The cleartext packet 902 is part of the program or instance of service 926 that is provided to the subscriber of DBDS 100 by an entitlement agent of the DBDS 100. The cleartext packet 902 can be stored at the headend 102 in storage devices such as VOD pump 226.

At the headend 102, the cleartext packet 902 is provided to a cryptographic device 518, which produces a packet of ciphertext 1006. The cryptographic device 518 is included in the multi-transport stream receiver-transmitter 228. In the preferred embodiment, the cryptographic device 518 employs a cryptographic algorithm such the 3DES cryptographic algorithm. The cryptographic device 518 receives the cleartext packet 902 and uses the keys k_1 1008, k_2 1010, and k_3 1012 to convert the cleartext

packet 902 into the ciphertext packet 1006. The keys k_1 1008, k_2 1010, and k_3 1012 are produced by the TED 302 that is associated with the entitlement agent providing the program or instance of service to the DBDS 100. The TED 302 sends the keys k_1 1008, k_2 1010, and k_3 1012 to the multi-transport stream receiver-transmitter 228 for use by the cryptographic device 518.

In an alternative embodiment, the cryptographic device 518 employs a multi-layered cryptographic algorithm such as DVB common scrambling or other multi-layer cryptographic algorithms known to those skilled in the art. In a multi-layer cryptographic algorithm the cleartext packet 902 is converted to the ciphertext packet 1006 by encrypting the cleartext multiple times using at least one key.

In the preferred embodiment, the encryption key k_1 1008 is a control word 524, which is frequently changed. Generally, multiple packets of cleartext 902 are encrypted by the cryptographic device 518 with a common control word such as $k_1(N)$ before the control word 1008 is changed to $k_1(N+1)$.

The packet of ciphertext 1006 is included in the transport stream 242 and transmitted to the subscriber's location 108. The keys 1008, 1010 and 1012 are included in ECMs or EMMs and sent to the DSCT 110. Typically, when multiple keys k_1 1008 are used to encrypt the program or instance of service 926, the keys are included in ECMs. However, in the case when key k_1 1008 is changed infrequently, or not at all during the transmission of the program or instance of service 926, the key(s) k_1 1008 can be included in either an ECM(s) or EMM(s). In an alternative embodiment, the headend 102 retains at least one of the keys 1008, 1010, or 1012 and provides the key in response to subscriber requests. By withholding one of the key(s) until receiving a request from the subscriber for the retained key(s), access to the content of the program or instance of service 926 is controlled by the operators of the DBDS 100. This enables the operator of the DBDS 100 to download and store the program or instance of service 926 at the subscriber's location while preventing the subscriber from making useable bootleg copies and from accessing the program or instance of service 926 until a predetermined time.

At the subscriber location 108, the cryptographic device 610 receives the ciphertext packet 1006 and uses a function of a cryptographic algorithm with the key k_3 1012 to convert the ciphertext packet 1006 to the ciphertext packet 1014. In the preferred embodiment, the decrypt function of the DES algorithm is used with the key k_3 1012 to remove a layer of encryption from the ciphertext packet 1006, and consequently, the ciphertext packet 1014 is written as $C''\{E, D; k_1, k_2\}$. The ciphertext

packet 1014 has two layers of encryption; it has been processed using an E function and a D function with keys k_1 1008 and k_2 1010, respectively.

The cryptographic device 610 reprocesses the ciphertext packet 1014 using an encryption (E) function of a cryptographic algorithm and the key k_4 1022 to further encrypt the ciphertext packet 1014, thereby producing the ciphertext packet 1020, which is written as $C'''\{E, D, E; k_1, k_2, k_4\}$. In the preferred embodiment, the cryptographic algorithm used for reprocessing the ciphertext packet 1014 is the DES cryptographic algorithm. The ciphertext packet 1020 corresponds to the cleartext packet 902 thrice encrypted by three different keys.

In the preferred embodiment, cryptographic device 518 of the multi-transport stream receiver-transmitter 228 converts the cleartext packet 902 to the ciphertext packet 1006 using the encryption function (E) of the 3DES algorithm with the keys k_1 1008, k_2 1010 and k_3 1012, and the cryptographic device 610 of the DSCT 110 converts the ciphertext packet 1006 to the ciphertext packet 1020 by applying the decryption function (D) and the encryption (E) function of the DES algorithm with the keys k_3 1012 and k_4 1022, respectively.

In the preferred embodiment, the key 1022 is generated locally in the DSCT 110 by the processor 608 or the secure processor 612. However, in an alternative embodiment, the key 1022 is generated at the headend 102 and sent to the DSCT 110 via EMMs or ECMs.

Ciphertext packet 1020 is stored in storage device 614, which is capable of storing multiple programs or instances of service 926 and associated information including media keys 924. Typically, storage device 614 is a hard-drive, CD, or other writeable electronic media known to those skilled in the art. In an alternative embodiment, the external storage device 650 stores the program or instance of service 926 and the media keys 924.

In the preferred embodiment, the storage device 614 stores all of the ciphertext packets 1020 of the program or instance of service 926 and all of the media keys 924, which are the keys used to decrypt the packets of the program or instance of service 926. The media keys 924 are then restricted to prevent the subscriber from providing unauthorized copies of the media keys 924 and program or instance of service 926. When the subscriber wishes to use the stored program or instance of service 926, the subscriber interacts with the DSCT 110 using a subscriber interface device (not shown) and requests the stored program or instance of service 926. The ciphertext packets 1020 of the stored program or instance of service 926 are sent to the cryptographic device 610 with their

appropriate media keys 924, i.e., keys k_1 1008, k_2 1010, and k_4 1022. The cryptographic device 610 uses the keys 1008, 1010, and 1022 with the appropriate encryption and decryption functions to convert the ciphertext 1020 to cleartext 902. As previously stated hereinabove, the key k_1 1008 can also be stored at the headend 102. The key k_1 1008 is then securely provided to the subscriber via EMMs or ECMs when the subscriber requests access to the stored program or instance of service.

In the preferred embodiment, the ciphertext packet 1020 corresponds to a packet of cleartext that has been converted to a ciphertext packet by 3DES encryption, and the cryptographic device 610 is adapted to convert the ciphertext packet 1020 to the cleartext packet 902 by using 3DES decryption and keys 1008, 1010 and 1022. Thus, the cryptographic device 518 and the cryptographic device 610 each perform at least one of the operations for converting cleartext packet 902 into a 3DES encrypted ciphertext packet 1020.

Generally, the cleartext 902 is received by the converter 618, which converts the cleartext from a packetized format to a format appropriate for user device 112. Typically, the user device is a television, computer, a VCR or other such device.

It is to be understood that this was just one example among many and that no particular significance should be placed upon the order in which the keys are used. For example, in an alternative embodiment, the key k_1 1008 can be a static key (i.e., it does not change during the transmission of the program or instance of service) and the key k_3 1112 can be a control word 524 that is frequently changed. In yet another alternative embodiment, the keys k_1 1008, k_2 1110 and k_3 1112 can be all static or all control words or any combination thereof.

EXAMPLE 3:

Referring to FIG. 11, which shows some of the functional components, among others, of the headend 102 and the DSCT 110, a packet of ciphertext 1102 is received at the headend 102 along with a key 1104. The ciphertext packet 1102 corresponds to a cleartext packet that was encrypted using a function of a cryptographic algorithm with the key 1104. In the preferred embodiment, the encrypt function (E) of the DES algorithm was used with the key 1104 to produce the ciphertext packet 1102. In an alternative embodiment, the decrypt function (D) is used with the key 1104 to produce the ciphertext packet 1102. The ciphertext packet 1102 is part of the program or instance of service 926 that is provided to the subscriber of DBDS 100 by the content provider 114, or

entitlement agent. The ciphertext packet 1102 and key 1104 can be stored at the headend 102 in storage devices such as VOD pump 226.

At the headend 102, the ciphertext packet 1102 is provided to the cryptographic device 518 of the multi-transport stream receiver-transmitter 228 along with an encryption key 1110. The cryptographic device 518 uses the encryption key 1110 and the decryption function (D) of a cryptographic algorithm to convert the ciphertext packet 1102 to the ciphertext C''{E, D; k_1, k_2} 1108, which has two layers of encryption.

In an alternative embodiment, the cryptographic device 518 uses the encryption key 1110 and the encrypt function (E) of a cryptographic algorithm to convert the ciphertext packet 1102 to the ciphertext packet 1108; and, the ciphertext packet 1108 is then written as C''{D, E; k_1, k_2} because it has two layers of encryption that were applied using the decrypt decryption (D) function with the key k_1 and the encrypt function (E) with the key k_2.

In the preferred embodiment, the encryption key k_2 1110 is a control word 524, which is frequently changed. Generally, the cryptographic device 518 uses a control word 524 such as $k_2(N)$ to encrypt multiple packets of a program or instance of service before the control word 524 is changed to $k_1(N+1)$. The ciphertext packet 1008 is included in the transport stream 242 (FIG. 2) and transmitted to the subscriber's location 108. The keys k_1 1104 and k_2 1110 are included in ECMs or EMMs and sent to the DSCT 110. Typically, when multiple keys k_2 1110 are used to encrypt the program or instance of service 926, the keys are included in ECMs. However, in the case when key k_2 1110 is changed infrequently, or not at all during the transmission of the program or instance of service, the key(s) k_2 1110 can be included in either an ECM(s) or EMM(s).

In an alternative embodiment, the headend 102 retains one or both of the keys k_1 1102 and k_2 1110 and provides the retained key, or keys, in response to subscriber requests. By withholding the key(s) until receiving a request for the key(s) from the subscriber, access to the content of the program or instance of service is controlled by the operators of the DBDS 100. This enables the operator of the DBDS 100 to download and store the programs or instances of service 926 at the subscriber's location while preventing the subscriber from making useable bootleg copies and from accessing the program or instance of service 926 until a predetermined time. The subscriber uses his or her user interface device (not shown) to request access to the stored program or instance of service 926, and the DSCT 110 then sends a request to the headend 102 for the key(s). The system controller 232 processes the request and sends the key(s) to the DSCT 110

and, if necessary, bills the subscriber for the program or instance of service. In an alternative embodiment, the entitlement agent that provides the program or instance of service to the DBDS 100 retains the key k_1 1102 until the subscriber requests the key.

At the subscriber location 108, the cryptographic device 610 receives the ciphertext packet 1108 and uses a function of a cryptographic algorithm with the key k_3 1114 to further encrypt the ciphertext packet 1108 . In the preferred embodiment, the encrypt function (E) of the DES cryptographic algorithm and the key k_3 1114 are used to convert the ciphertext packet 1108 to the ciphertext packet 1116. The ciphertext packet 1116, which is the result of adding a layer of encryption to the ciphertext packet 1108, is written as $C'''\{E, D, E; k_1, k_2, k_3\}$. The ciphertext packet 1116 has three layers of encryption; it has been encrypted using an E function, a D function and an E function with keys k_1, k_2 and k_3, respectively.

In the preferred embodiment, the cryptographic algorithm used for processing the ciphertext packet 1108 is the DES cryptographic algorithm. The ciphertext packet 1116 corresponds to a cleartext packet thrice encrypted by three different keys. In the preferred embodiment, cryptographic device 518 of the multi-transport stream receiver-transmitter 228 receives the ciphertext packet 1102, which has one layer of encryption that was applied by the encryption function (E) of the DES algorithm with the key k_1 1104, and the cryptographic device 518 converts the ciphertext packet 1102 to the ciphertext packet 1108 using the decrypt function (D) of the DES algorithm with the key k_2. The cryptographic device 610 of the DSCT 110 applies a third layer of encryption using the encryption (E) function of the DES algorithm with the key k_3 1114 to produce the ciphertext packet 1116. The ciphertext packet 1116 corresponds to encrypting the cleartext packet 902 using the encryption (E) function of 3DES cryptographic algorithm with the keys k_1 1102, k_2 1110 and k_3 1114.

In the preferred embodiment, the key 1114, which is used by the cryptographic device 610 for encrypting, is preferably generated locally in the DSCT 110 by the processor 608 or the secure processor 612. However, in an alternative embodiment, the key 1114 is generated at the headend 102 and sent to the DSCT 110 via EMMs or ECMs.

In the preferred embodiment, the ciphertext packet 1116 is stored in the storage device 614. Typically, storage device 614 is a hard-drive, CD, or other writeable electronic media known to those skilled in the art, which is capable of storing multiple programs or instances of service and associated information. In an alternative

embodiment, the ciphertext packet 1116 is stored in the external storage device 650 (FIG. 6).

In the preferred embodiment, the storage device 614 stores all of the ciphertext packets 1116 of the program or instance of service 926 and the media keys 924, which are the keys used for decrypting the ciphertext packet 1116 of the program or instance of service 926. The media keys 924 are then restricted to prevent the subscriber from providing unauthorized copies of the media keys 924 and the program or instance of service 926. When the subscriber wishes to use the stored program or instance of service 926, the subscriber interacts with the DSCT 110 using a subscriber interface device (not shown) such as a remote control and requests the stored program or instance of service 926. The ciphertext packets 1116 of the stored program or instance of service 926 are sent to the cryptographic device 610 along with the appropriate media keys 924, i.e., keys k_1 1104, k_2 1110, and k_3 1114. The cryptographic device 610 uses the keys 1104, 1110 and 1114 with the appropriate encryption and decryption functions of the cryptographic algorithm to convert the ciphertext 1116 to the cleartext 902.

As previously stated hereinabove, the keys 1104 and 1110 can also be stored at the headend 102. In that case, the keys 1104 and 1110 are then securely provided to the subscriber via EMMs or ECMs when the subscriber requests access to the stored program or instance of service, and it is provided to the cryptographic device 610.

In the preferred embodiment, the cryptographic device 518 and the cryptographic device 610 each perform at least one of the operations for converting the ciphertext packet 1102 into a 3DES encrypted ciphertext packet 1116, and the cryptographic device 610 converts the ciphertext packet 1116 to the cleartext packet 902 using the decryption function of the 3DES cryptographic algorithm with the keys 1104, 1110 and 1114.

Generally, the cleartext 902 is received by the converter 618, which converts the cleartext from a packetized format to a format appropriate for user device 112. Typically, the user device 112 is a television, a computer, a VCR or other such device.

It should be emphasized that the above-described embodiments and examples of the present invention, particularly, any "preferred" embodiments, are merely possible examples of implementations, merely set forth for a clear understanding of the principles of the invention. Many variations and modifications may be made to the above-described embodiment(s) of the invention without departing substantially from the spirit and principles of the invention. All such modifications and variations are intended to be

included herein within the scope of this disclosure and the present invention and protected by the following claims.